Expert Systems with Applications 37 (2010) 2826–2837

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Self-adaptive harmony search algorithm for optimization

Chia-Ming Wang[a], Yin-Fu Huang[b,*]

[a] Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology, 123 University Road, Section 3, Touliou, Yunlin 640, Taiwan, ROC
[b] Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, 123 University Road, Section 3, Touliou, Yunlin 640, Taiwan, ROC

ARTICLE INFO

Keywords:
Harmony search
Low-discrepancy sequence
Meta-heuristic algorithm
Optimization

ABSTRACT

Recently, a new meta-heuristic optimization algorithm – harmony search (HS) with continuous design variables was developed. This algorithm is conceptualized using the musical improvisation process of searching for a perfect state of harmony. Although several variants and an increasing number of applications have appeared, one of its main difficulties is how to select suitable parameter values. In this paper, we used the consciousness (i.e., harmony memory) to automatically adjust parameter values. In addition, the pseudo-random number generator is also replaced by the low-discrepancy sequences for initialization of the harmony memory. Finally, the experimental results revealed the superiority of the proposed method to the original HS and recently developed variants.

1. Introduction

Owing to a variety of complex optimization problems in a broad field having been confronted successfully with meta-heuristics approaches (Pardalos & Resende, 2002), there has been an increasing interest in the study of brand new or hybrid algorithms during the last few years. Another motivation that encourages researchers might be that the meta-heuristics algorithms are general purpose optimizers that do not require special conditions or mathematical properties of the objective functions, such as gradient information and differentiable properties.

In recent years, Geem, Kim, and Loganathan (2001), Lee and Geem (2005) proposed a new meta-heuristic algorithm, harmony search (HS), which imitates the behaviors of music improvisation process. Similarly, the HS has been successfully applied to several real-world optimization problems (Geem, 2008). Nevertheless, it also suffers from a serious problem as other meta-heuristics does; its capabilities are quite sensitive to parameter setting. Even though a few of previous studies were concerned with parameter setting (Geem, 2007; Omran & Mahdavi, 2008), those experiments were not rigorous enough (e.g. no replications). Furthermore, existing conclusions and heuristic values of parameters are often obtained from low-dimensional problems (usually less than 10 variables) and one-factor-at-a-time experiments.[1] Thus, researchers and practitioners might get wrong impressions on the HS and ignore possible interactions of control parameters.

In this paper, we proposed a new harmony search variant, whose parameters are automatically adjusted according to its self-consciousness. In addition, we carefully conducted a set of experiments to reveal the impact of control parameters. Accordingly, there is no need to tune the control parameters, thereby leading an almost-parameter-free harmony search algorithm.

The remainder of this paper is organized as follows: in Section 2, a general harmony search algorithm and its recently developed variants would be reviewed. Section 3 introduced our method that could automatically adjust parameters, and initialize the harmony memory with the low-discrepancy sequences as well. Section 4 presented the empirical results and discussions. Finally, conclusions were given in the last section.

2. Previous work

In this section, we brief review the harmony search algorithm and recently developed variants.

2.1. Original harmony search

The general procedures of harmony search are as follows (Geem et al., 2001):

Step 1 Create and randomly initialize a harmony memory (HM) with size HMS.
Step 2 Improvise a new harmony from the HM.

* Corresponding author. Tel.: +886 5 5342601x4314; fax: +886 5 5312063.
E-mail addresses: cmwang@ieee.org (C.-M. Wang), huangyf@yuntech.edu.tw (Y.-F. Huang).

[1] One-factor-at-a-time experiments are varied with only one factor or variable at a time while keeping others fixed (Czitrom, 1999).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.09.008

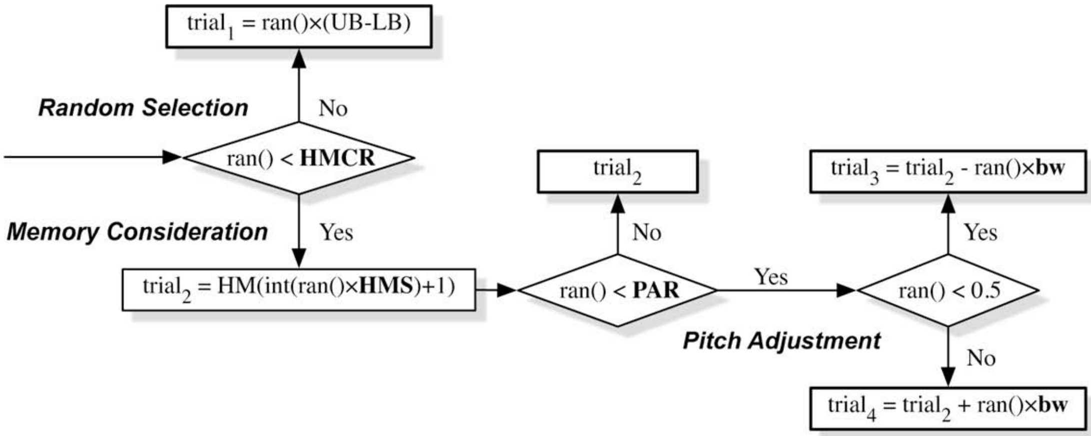


Fig. 1. A new harmony improvisation process, where HMS is the memory size, HMCR is the rate of choosing from the memory, PAR is a pitch adjustment rate, and bw is the distance bandwidth.

Step 3 Update the HM. If the new harmony is better than the worst harmony in the HM, include the new harmony into the HM, and exclude the worst harmony from the HM.

Step 4 Repeat Steps 2 and 3 until the maximum number of iterations is reached.

A harmony memory (HM) is a set of solution vectors, and it is convenient to view the HM as a population of a genetic algorithm. In fact, the harmony search is completely governed by three rules in Step 2 – random selection, memory consideration and pitch adjustment, as shown in Fig. 1.

The parameter HMCR, which varies between 0 and 1, controls the balance between exploration and exploitation. For example, an HMCR value 0 behaves like a purely random search; i.e., randomly selecting a value from the possible range of variables. On the contrary, there is no chance to improve the harmony from outside the HM, so that it merely randomly picks one pitch from the HM. Once a pitch has been picked from the HM, the pitch adjustment rate (PAR) determines whether further adjustment is required according to a variable distance bandwidth (bw). In other words, the pitch adjustment step is similar to the local search mechanism, and the variable distance bandwidth is its step size. Consequently, it shows clearly that PAR and bw have a great influence on the quality of final solutions.

2.2. Improved harmony search

Because PAR and bw in the HS algorithm control the convergence rate and the ability for fine-tuning, Mahdavi, Fesanghary, and Damangir (2007) proposed a variant of HS, called the improved harmony search (IHS), to dynamically increase PAR and decrease bw, respectively. The IHS gets rid of the weaknesses based on fixed values of PAR and bw in the HS algorithm.

Indubitably, bw has a considerable influence on the precision of solutions and should be problem dependent. Therefore, it seems reasonable that decreasing bw with an iteration number[2] could fine-tune the final solutions. This philosophy is the same as dynamically decreasing the learning rate of neural networks (Haykin, 2007). However, how to decide a suitable set of bw_{min} and bw_{max} becomes another new problem.

On the other hand, increasing PAR continuously in the HS might be a questionable strategy for some reasons. Since PAR controls the probability to either simply select a pitch from the harmony memory randomly or further adjust a selected pitch, we believe that a successful search should be proceeded progressively in the beginning, and then gradually settled down. Thus, PAR should be decreased with time to prevent overshooting and oscillation. Besides, this questionable strategy is apparently contradicted to the previous statements about bw. It is also interesting to note that the global-best harmony search (Omran & Mahdavi, 2008) with relatively small constant PAR instead of a variable, proposed by the same authors, can obtain better results!

2.3. Global-best harmony search

A recently developed variant of HS, called the global-best harmony search (GHS) (Omran & Mahdavi, 2008), which borrowed the concepts from swarm intelligence to enhance its performance. The GHS directly adopts the current best pitch from the harmony memory to simplify the pitch adjustment step, thereby eliminating the difficulties in selecting bw. Although this variant sounds great due to its name, there are a number of problems with it.

Because a harmony search belongs to neighborhood meta-heuristics[3], it would merely use its own past experiences. For that reason, there are neither swarms nor relative global concepts in the HS. The term "global-best" not only seems misused, but also confuses other researchers. Additionally, although selecting the current best pitch from the harmony memory can keep away from bw, it also causes a serious side effect – premature convergence (to be discussed later). Moreover, there are some obvious mistakes in the GHS, so that the reliability of the numerical results is decreased.

3. Proposed method

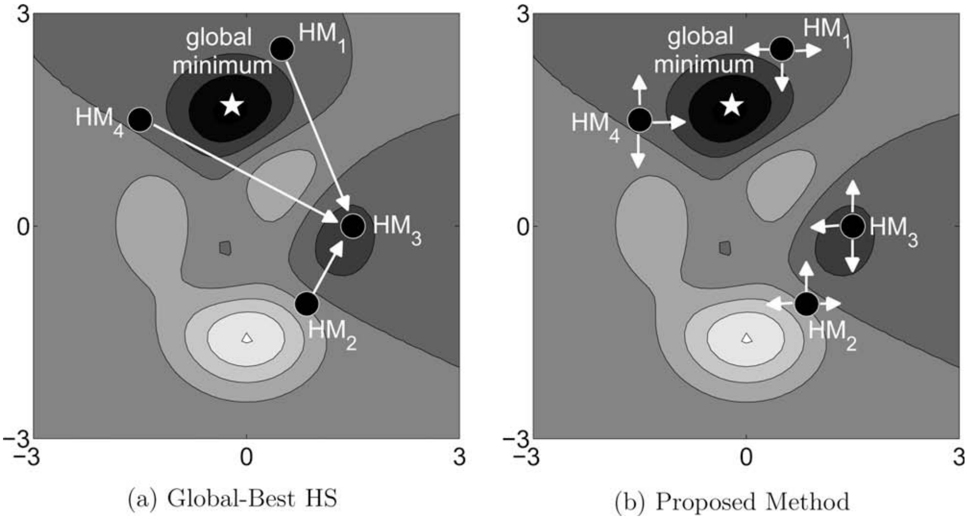
In this section, we introduced our method and the low-discrepancy sequences for initialization of the harmony memory.

3.1. Self-adaptive mechanism

As stated before, PAR and bw may somewhat influence the quality of final solutions, so we modified the pitch adjustment step of the HS such that the new harmony can better utilize its own

[2] $bw(iter) = bw_{max} exp\left(\frac{\ln(bw_{min}) - \ln(bw_{max})}{MaxIter} \times iter\right)$, where bw is decreased exponentially. It is a common power law function.

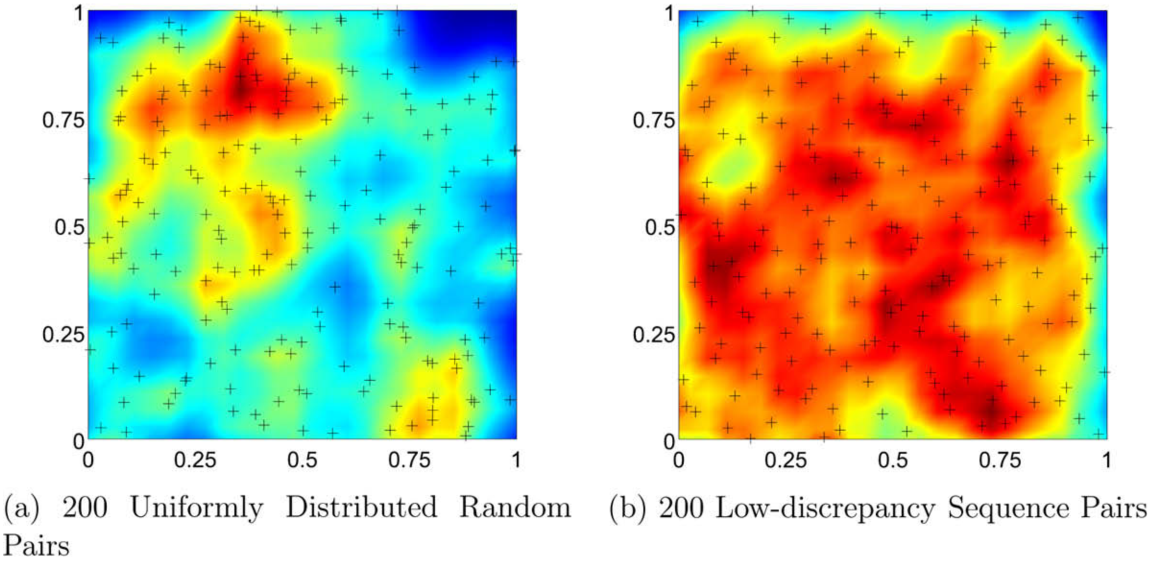
[3] There are two major types of meta-heuristics; one is population-based and another is neighborhood. Neighborhood meta-heuristics such as simulated annealing (Glover, 1989) and tabu search (Kirkpatrick, Gelatt, & Vecchi, 1983) only evaluate one potential solution at a time. They are very different from population-based heuristics, where a set of potential solutions simultaneously move toward goals.



(a) Global-Best HS

(b) Proposed Method

Fig. 2. Conceptual diagrams of the pitch adjustment mechanism, where HMS = 4, HMCR = 1, PAR = 1, and $f(\mathrm{HM}_3) < \{f(\mathrm{HM}_1), f(\mathrm{HM}_2), f(\mathrm{HM}_4)\}$.



(a) 200 Uniformly Distributed Random Pairs

(b) 200 Low-discrepancy Sequence Pairs

Fig. 3. Uniformly distributed random numbers and low-discrepancy sequences.

Table 1
Benchmark functions.

Function	Function range	x^*	$f_{\min}$
Sphere $= \sum_{i=1}^{n} x_i^2$	$-100 \leqslant x_i \leqslant 100$	$[0, \ldots, 0]$	0
Rosenbrock $= \sum_{i=1}^{n-1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2$	$-2.048 \leqslant x_i \leqslant 2.048$	$[1, \ldots, 1]$	0
Ackley $= -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + \exp(1)$	$-32.768 \leqslant x_i \leqslant 32.768$	$[0, \ldots, 0]$	0
Griewank $= \sum_{i=1}^{n} \frac{(x_i-100)^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i-100}{\sqrt{i}}\right) + 1$	$-600 \leqslant x_i \leqslant 600$	$[100, \ldots, 100]$	0

experiences. More precisely, we replaced the parameter bw altogether and updated the new harmony according to the maximal and minimal values in the HM. Let $min(HM^i)$ and $max(HM^i)$, respectively denote the lowest and the highest values of the ith variable in the HM, and then the selected pitch from the HM called a trial is further adjusted by the following equations:

$$trial^i + [max(HM^i) - trial^i] \times ran[0,1) \tag{1}$$

$$trial^i - [trial^i - min(HM^i)] \times ran[0,1) \tag{2}$$

where $ran[0,1)$ is a uniform number in the $[0,1]$ range without 1.[4] Intuitively, the $min(HM^i)$ and $max(HM^i)$ would approach the optimum gradually, and thus this mechanism could progressively make finer adjustments to the harmony. Besides, it is clear that the pitch adjustment through this mechanism would not violent the boundary constraint of variables.

[4] Although we use a half-closed interval $[0,1)$ here, it can be exchanged by $[0,1]$ seamlessly.

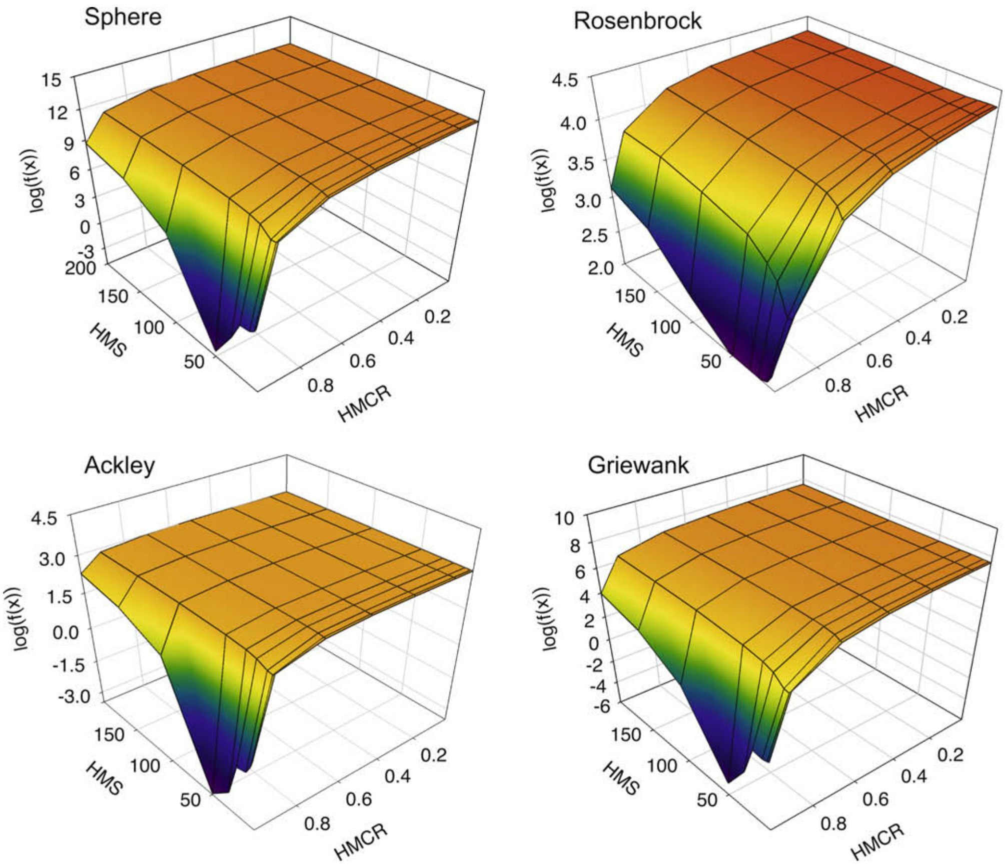


Fig. 4. Full-factorial designed experiments of HMCR and HMS, Dim = **100**. All results have been averaged over 50 runs.

Table 2
Default parameters for all methods, where the HS$_{50}$, IHS$_{50}$ and GHS$_{50}$ are initialized with the same HMS and HMCR as the proposed method.

Methods	HMS	HMCR	PAR	PAR$_{min}$	PAR$_{max}$	bw	bw$_{min}$	bw$_{max}$
HS	5	0.9	0.3	–	–	0.01	–	–
IHS	5	0.9	–	0.01	0.99	–	0.0001	1/(20 × (UB–LB))
GHS	5	0.9	–	0.01	0.99	–	–	–
HS$_{50}$	50	0.99	0.3	–	–	0.01	–	–
IHS$_{50}$	50	0.99	–	0.01	0.99	-	0.0001	1/(20 × (UB–LB))
GHS$_{50}$	50	0.99	–	0.01	0.99	–	–	–
Our method	50	0.99	–	0	1	–	–	–

Table 3
Initialization ranges for experiments.

Function	Symmetric		Positive symmetric		Negative symmetric	
	Min	Max	Min	Max	Min	Max
Sphere	−100	100	50	100	−100	−50
Rosenbrock	−2.048	2.048	1.024	2.048	−2.048	−1.024
Ackley	−32.768	32.768	16.384	32.768	−32.768	−16.384
Griewank	−600	600	300	600	−600	−300

Comparing it with the GHS in Fig. 2 where the memory size is 4, and the other two rules are disabled to simplify our discussion. All the harmonies in the left figure have a tendency toward the current best harmony HM$_3$, despite its actual quality, and this may lead to premature convergence. On the contrary, all the harmonies in the right figure have a chance to adjust their positions between the maximal and minimal values of corresponding variables. As searching goes on, they would eventually settle down near the global optimum. In addition, we 'decrease' the PAR linearly with time, the reasons have been explained in Section 2.2.

3.2. Initialization with the low-discrepancy sequences

If the harmonies can be spread out uniformly in the search space, it is much more likely to converge to a better solution. Therefore, we use the low-discrepancy sequences (Lecot, 1989)

Table 4
Mean and standard deviation of the benchmark function optimization results in **30** dimensions. The results were averaged over 30 runs. The last column indicates the numbers of significance at $\alpha = 0.01$ between our method and others by two-tailed *t*-test.

	HS	HS$_{50}$	IHS	IHS$_{50}$	GHS	GHS$_{50}$	Ours	*t*-test
Symmetric								
Sphere	2.7839E+01	1.5080E+01	2.8123E+01	1.3910E+01	3.0314E+01	1.4226E+01	6.9160E−07	(6/6)
	(8.5171E+00)	(4.2610E+00)	(7.2525E+00)	(3.3366E+00)	(1.3092E+01)	(4.2203E+00)	(1.1035E−06)	–
Rosenbrock	6.9036E+01	6.7302E+01	8.6569E+01	6.5892E+01	8.3760E+01	6.4184E+01	2.6468E+01	(6/6)
	(2.5807E+01)	(2.4564E+01)	(1.8392E+01)	(2.4328E+01)	(1.7622E+01)	(3.1200E+01)	(5.6833E−01)	–
Ackley	2.1430E+00	2.0563E+00	2.6719E+00	2.0589E+00	2.6267E+00	1.9627E+00	7.8081E−04	(6/6)
	(2.2839E−01)	(2.9215E−01)	(1.5991E−01)	(2.4955E−01)	(2.3225E−01)	(2.3377E−01)	(4.6613E−04)	–
Griewank	1.2626E+00	1.1273E+00	1.2590E+00	1.1273E+00	1.2594E+00	1.1300E+00	8.4515E−05	(6/6)
	(6.2004E−02)	(3.5265E−02)	(8.4917E−02)	(3.9869E−02)	(7.9548E−02)	(3.2676E−02)	(2.3783E−04)	–
Pos-Asy.								
Sphere	2.8659E+01	1.7396E+01	3.3924E+01	1.5275E+01	2.9558E+01	1.3932E+01	7.3453E−07	(6/6)
	(7.8289E+00)	(7.3148E+00)	(9.2825E+00)	(4.3709E+00)	(7.2631E+00)	(5.2507E+00)	(1.0640E−06)	–
Rosenbrock	6.7621E+01	6.4767E+01	8.8567E+01	7.1195E+01	8.0879E+01	7.4409E+01	2.6227E+01	(6/6)
	(2.4265E+01)	(2.3599E+01)	(1.2487E+01)	(2.2561E+01)	(3.0205E+01)	(3.2250E+01)	(7.7277E−01)	–
Ackley	2.0397E+00	2.0186E+00	2.6205E+00	1.9979E+00	2.6282E+00	1.9460E+00	7.3474E−04	(6/6)
	(1.7113E−01)	(2.6183E−01)	(2.5072E−01)	(2.4852E−01)	(2.3894E−01)	(2.9917E−01)	(4.9978E−04)	–
Griewank	1.2607E+00	1.1275E+00	1.2816E+00	1.1380E+00	1.2968E+00	1.1391E+00	2.8618E−06	(6/6)
	(7.9909E−02)	(4.7671E−02)	(9.1823E−02)	(3.7465E−02)	(1.1601E−01)	(4.7084E−02)	(5.6166E−06)	–
Neg-Asy.								
Sphere	2.6991E+01	1.4283E+01	2.8015E+01	1.4800E+01	2.8738E+01	1.3522E+01	8.6273E−07	(6/6)
	(6.9635E+00)	(4.5010E+00)	(7.7629E+00)	(3.9032E+00)	(6.4934E+00)	(4.1964E+00)	(9.6601E−07)	–
Rosenbrock	6.9996E+01	7.3252E+01	8.5577E+01	6.3739E+01	9.2027E+01	7.7417E+01	2.6325E+01	(6/6)
	(1.9284E+01)	(2.7232E+01)	(1.7473E+01)	(2.8103E+01)	(3.1433E+01)	(3.4397E+01)	(6.5309E−01)	–
Ackley	2.0604E+00	2.0043E+00	2.6756E+00	1.9139E+00	2.5796E+00	2.0236E+00	8.8612E−04	(6/6)
	(1.6119E−01)	(2.8361E−01)	(1.9324E−01)	(2.3872E−01)	(1.8703E−01)	(2.0861E−01)	(6.3377E−04)	–
Griewank	1.2872E+00	1.1363E+00	1.2703E+00	1.1340E+00	1.2472E+00	1.1393E+00	1.1649E−05	(6/6)
	(8.5598E−02)	(4.7167E−02)	(1.0100E−01)	(3.8907E−02)	(6.5667E−02)	(3.8543E−02)	(2.2671E−05)	–

Table 5
Mean and standard deviation of the benchmark function optimization results in **100** dimensions. The results were averaged over 30 runs. The last column indicates the numbers of significance at $\alpha = 0.01$ between our method and others by two-tailed *t*-test.

	HS	HS$_{50}$	IHS	IHS$_{50}$	GHS	GHS$_{50}$	Ours	*t*-test
Symmetric								
Sphere	1.4496E+04	2.1740E+01	1.5373E+04	2.4987E+01	1.6361E+04	2.0145E+01	1.5321E−02	(6/6)
	(1.2675E+03)	(3.8355E+00)	(1.2652E+03)	(5.1797E+00)	(1.0244E+03)	(3.0045E+00)	(1.1839E−02)	–
Rosenbrock	1.2241E+03	2.4755E+02	1.1552E+03	2.6187E+02	1.1962E+03	2.6110E+02	9.5993E+01	(6/6)
	(1.2221E+02)	(5.0723E+01)	(1.0682E+02)	(4.5400E+01)	(9.5991E+01)	(4.2179E+01)	(5.4020E−01)	–
halfline								
Ackley	1.2487E+01	1.2294E+00	1.2607E+01	1.1956E+00	1.2801E+01	1.2179E+00	2.9248E−02	(6/6)
	(2.2604E−01)	(1.3223E−01)	(3.0371E−01)	(9.6269E−02)	(2.3092E−01)	(1.3273E−01)	(9.8086E−03)	–
Griewank	1.4061E+02	1.2050E+00	1.4207E+02	1.2107E+00	1.4635E+02	1.1687E+00	6.6646E−03	(6/6)
	(1.0843E+01)	(3.4919E−02)	(1.2511E+01)	(3.2800E−02)	(1.3283E+01)	(3.3162E−02)	(3.8819E−03)	–
Pos-Asy.								
Sphere	1.4920E+04	2.0737E+01	1.5441E+04	2.2659E+01	1.6590E+04	2.0800E+01	1.1234E−02	(6/6)
	(1.2797E+03)	(3.0322E+00)	(1.6554E+03)	(3.0208E+00)	(1.5075E+03)	(3.5454E+00)	(9.0235E−03)	–
Rosenbrock	1.1830E+03	2.4551E+02	1.1463E+03	2.2606E+02	1.2071E+03	2.5063E+02	9.6320E+01	(6/6)
	(1.1440E+02)	(3.3002E+01)	(1.0379E+02)	(3.9461E+01)	(9.6643E+01)	(3.9527E+01)	(5.9583E−01)	–
Ackley	1.2543E+01	1.2489E+00	1.2669E+01	1.2100E+00	1.2862E+01	1.2192E+00	2.6441E−02	(6/6)
	(2.7652E−01)	(1.0997E−01)	(2.9887E−01)	(1.7145E−01)	(2.9501E−01)	(1.2295E−01)	(6.7872E−03)	–
Griewank	1.4034E+02	1.2044E+00	1.4087E+02	1.2115E+00	1.4812E+02	1.1700E+00	4.0883E−03	(6/6)
	(1.3586E+01)	(3.3912E−02)	(8.4121E+00)	(3.1230E−02)	(1.3518E+01)	(2.4150E−02)	(2.4742E−03)	–
Neg-Asy.								
Sphere	1.5005E+04	2.1909E+01	1.5495E+04	2.1822E+01	1.6212E+04	1.9844E+01	1.2420E−02	(6/6)
	(1.0968E+03)	(3.4901E+00)	(1.4779E+03)	(3.8574E+00)	(1.1895E+03)	(3.2543E+00)	(1.1817E−02)	–
Rosenbrock	1.2147E+03	2.4272E+02	1.1747E+03	2.4062E+02	1.1780E+03	2.5337E+02	9.6473E+01	(6/6)
	(1.1992E+02)	(4.0825E+01)	(1.0612E+02)	(4.6921E+01)	(1.0655E+02)	(3.6112E+01)	(5.2266E−01)	–
Ackley	1.2533E+01	1.2012E+00	1.2537E+01	1.1945E+00	1.2840E+01	1.2725E+00	2.8970E−02	(6/6)
	(2.9300E−01)	(1.1066E−01)	(3.0133E−01)	(9.9816E−02)	(2.2510E−01)	(1.3915E−01)	(6.1588E−03)	–
Griewank	1.4903E+02	1.2141E+00	1.4157E+02	1.2041E+00	1.4549E+02	1.1887E+00	9.7983E−03	(6/6)
	(1.2065E+01)	(3.3635E−02)	(1.2213E+01)	(2.4172E−02)	(1.2958E+01)	(3.2901E−02)	(9.7899E−03)	–

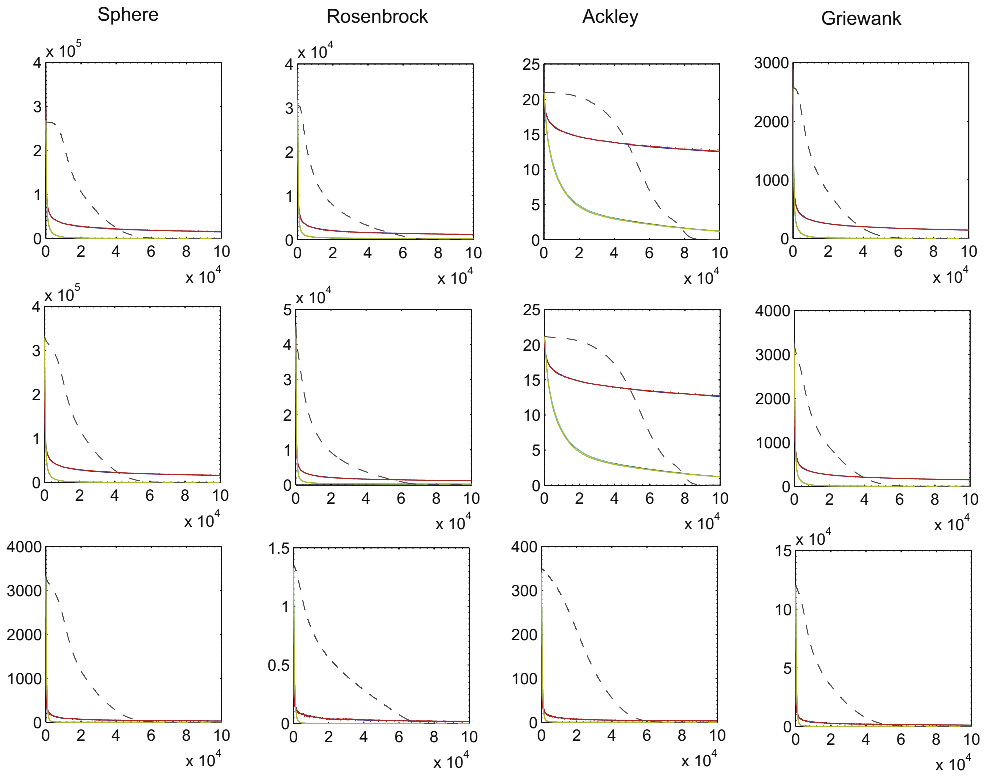
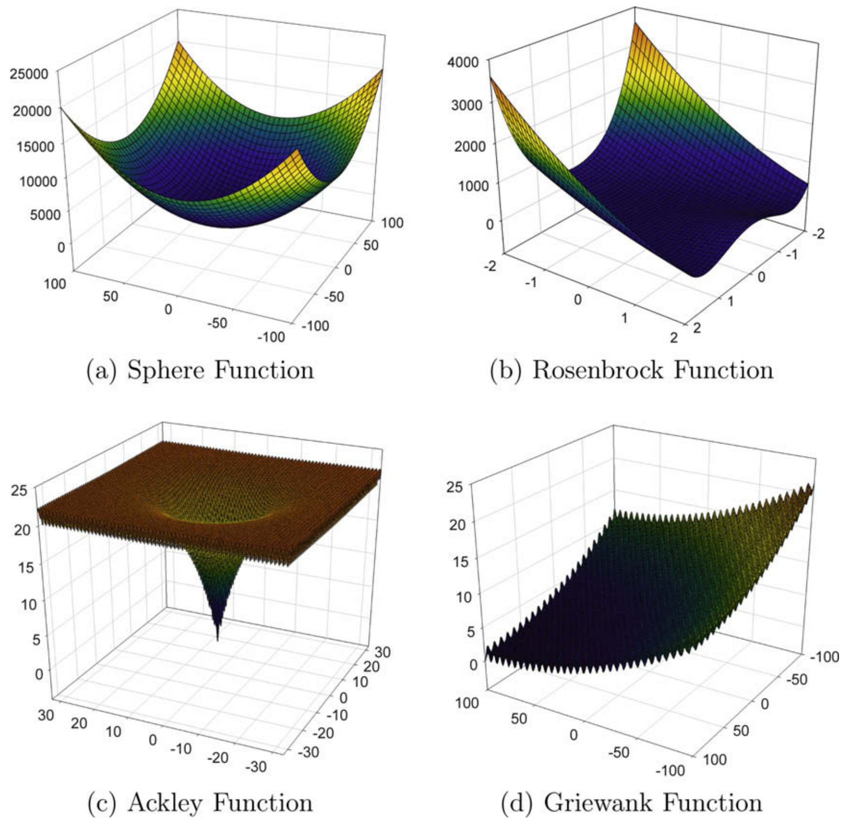


Fig. 5. Comparisons among seven methods with **symmetric** initial range on four test functions in **100** dimensions. The dash lines indicate the results of the proposed method. The apparent red lines indicate the results of the original HS, IHS and GHS, and the green lines indicate the HS_{50}, IHS_{50} and GHS_{50}. They are too close to be distinct. Three rows respectively indicate the best result, the average result, and the diversity. The vertical axes of the first two rows and the third row are the function values and the diversity, respectively. The horizontal axis is the number of function evaluations. All results have been averaged over 30 runs.



(a) Sphere Function

(b) Rosenbrock Function

(c) Ackley Function

(d) Griewank Function

Fig. 6. 2 D surfaces of test functions

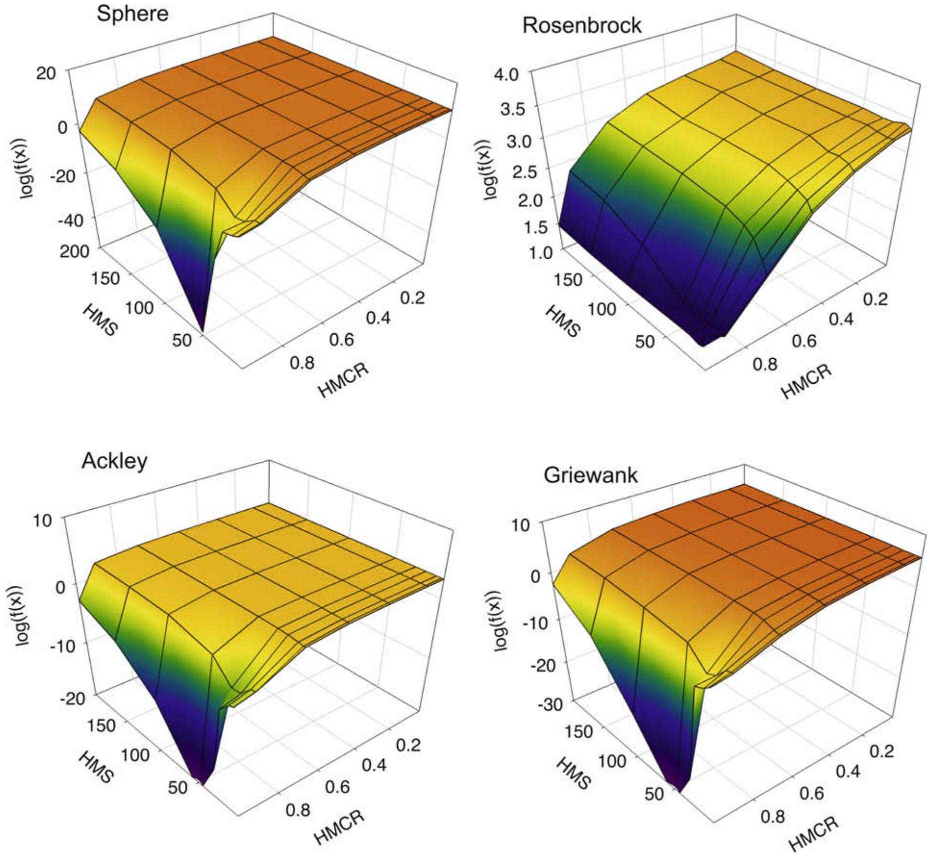


Fig. 7. Full-factorial designed experiments of HMCR and HMS, Dim = **30**. All results have been averaged over 50 runs.

Table 6
The *p*-values of the two-tailed *t*-test between our method and others in **30 dimension**. The results were averaged over 30 runs.

	HS	HS_{50}	IHS	IHS_{50}	GHS	GHS_{50}
Symmetric						
Sphere	3.2303E−17	3.8285E−18	3.1999E−19	4.3924E−20	2.3398E−13	1.4179E−17
Rosenbrock	6.2439E−10	5.2750E−10	3.1505E−17	9.1461E−10	3.5985E−17	2.9430E−07
Ackley	4.9400E−30	1.8338E−26	2.9539E−37	1.9701E−28	2.2961E−32	1.1950E−28
Griewank	9.6199E−40	2.0410E−45	9.2651E−36	7.1500E−44	1.3926E−36	2.0839E−46
Pos-Asy.						
Sphere	1.5341E−18	1.2043E−13	1.6041E−18	5.3723E−18	8.5214E−20	7.5778E−15
Rosenbrock	2.9840E−10	7.7133E−10	2.3986E−22	8.5648E−12	8.0411E−11	5.0367E−09
Ackley	5.0770E−33	1.3667E−27	2.2209E−31	4.1375E−28	5.1102E−32	1.7255E−25
Griewank	1.5399E−36	1.2627E−41	5.2996E−35	9.0001E−45	3.1917E−32	6.5546E−42
Neg-Asy.						
Sphere	3.2354E−19	7.0979E−17	2.2567E−18	5.8945E−19	8.4402E−21	4.7238E−17
Rosenbrock	3.9576E−13	2.3973E−10	1.1496E−17	4.9527E−08	2.7994E−12	5.6821E−09
Ackley	6.7701E−34	1.6363E−26	6.7336E−35	4.4744E−28	7.5300E−35	1.9042E−30
Griewank	6.1617E−36	7.4059E−42	1.0666E−33	2.9773E−44	7.2148E−39	1.9789E−44

instead of the pseudo-random number[5] to initialize the harmony memory. The reason can be easily realized from Fig. 3.

Note the obvious differences between the random data and the low-discrepancy sequences. This is exactly how a uniformly-distributed random generator is expected to behave, since it generates the random numbers independently. In contrast, the low-discrepancy sequences are much more evenly distributed although there are some correlations between variables. However, this is the distribution we desired.

4. Empirical results

In this section, four common numerical optimization problems were chosen to compare the performance of our method against the original harmony search (HS), the improved harmony search (IHS), and the global-best harmony search (GHS).

4.1. Standard test functions

The four numerical optimization problems used in our empirical studies are the unimodal Sphere and Rosenbrock functions, and the multimodal Ackley and Griewank functions. All functions

[5] We used the famous Mersenne Twister (MT19937) (Matsumoto & Nishimura, 1998) as a pseudo-random number generator in this work.

Table 7
The *p*-values of the two-tailed *t*-test between our method and others in **100 dimension**. The results were averaged over 30 runs.

	HS	HS_{50}	IHS	IHS_{50}	GHS	GHS_{50}
Symmetric						
Sphere	1.6542E−32	8.5389E−24	2.8863E−33	7.8362E−22	1.0802E−36	7.3672E−26
Rosenbrock	7.7982E−30	3.4624E−16	9.9802E−31	1.6062E−18	1.5450E−32	2.4572E−19
Ackley	1.9902E−52	7.8589E−30	9.4161E−49	1.0753E−33	1.8224E−52	1.1638E−29
Griewank	4.4317E−34	3.4987E−47	2.0304E−32	3.6086E−48	4.8442E−32	1.5103E−47
Pos-Asy.						
Sphere	9.4889E−33	4.1760E−26	5.7852E−30	2.9988E−27	5.0012E−32	3.2196E−24
Rosenbrock	3.4312E−30	4.6108E−21	5.6134E−31	2.7389E−17	1.4266E−32	2.6360E−19
Ackley	7.7079E−50	2.5707E−32	5.5785E−49	2.8172E−26	2.4585E−49	1.4223E−30
Griewank	3.0935E−31	4.4414E−47	2.7488E−37	2.8502E−48	5.6721E−32	1.9435E−51
Neg-Asy.						
Sphere	9.4377E−35	4.7495E−25	2.0149E−31	8.9764E−24	1.0530E−34	1.0822E−24
Rosenbrock	5.8275E−30	2.7392E−18	4.9567E−31	1.6670E−16	5.1031E−31	1.3939E−20
Ackley	4.3787E−49	1.1126E−31	9.8231E−49	6.1775E−33	9.7353E−53	1.6700E−29
Griewank	1.8032E−33	9.4274E−53	1.1208E−32	3.7473E−63	2.8158E−32	5.0948E−53

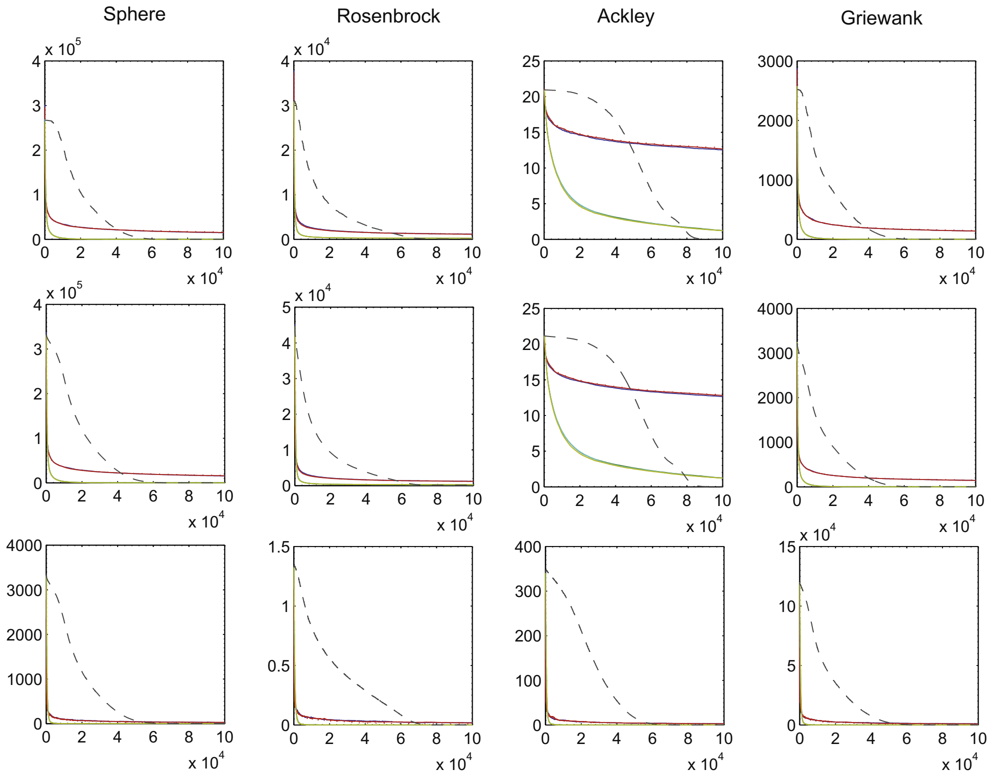


Fig. 8. Comparisons among seven methods with **positive symmetric** initial range on four test functions in **100** dimensions. The dash lines indicate the results of the proposed method. The apparent red lines indicate the results of the original HS, IHS and GHS, and the green lines indicate the HS_{50}, IHS_{50} and GHS_{50}. They are too close to be distinct. Three rows respectively indicate the best result, the average result, and the diversity. The vertical axes of the first two rows and the third row are the function values and the diversity, respectively. The horizontal axis is the number of function evaluations. All results have been averaged over 30 runs.

were implemented in 30 and 100 dimensions. Table 1 summarizes the information of these test functions.

4.2. Design of experiments

In order to reveal the impact of control parameters, we conducted full-factorial experiments[6] (Montgomery & Runger, 2006) for our method, where HMS = 3, 5, 10, 20, 30, 50, 100, 150, 200 (9 levels) and HMCR = 0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99 (7 levels). We conducted two sets of experiments on each test functions in 30 and 100 dimensions, and the results are presented in Fig. 4 and Fig. 7, respectively. All results have been averaged over 50 runs, and each run was allowed to run for 100,000 evaluations of the objective functions in 100 dimensions so as to preserve convergence. It is obvious from Fig. 4 that increasing the HMCR value improves the performance for all functions, and an appropriate value of the HMS is around 50. Similar results were also obtained in 30 dimensions (see Fig. 7).

4.3. Experimental settings

For all tests presented in this paper, the default parameters of all methods are listed in Table 2. These values were suggested by their authors Lee and Geem (2005), Mahdavi et al. (2007), Omran

[6] Some might argue, of course, that the fractional factorial experiment is a much more experimentally efficient method, but there is no need to save little time for computer simulations.

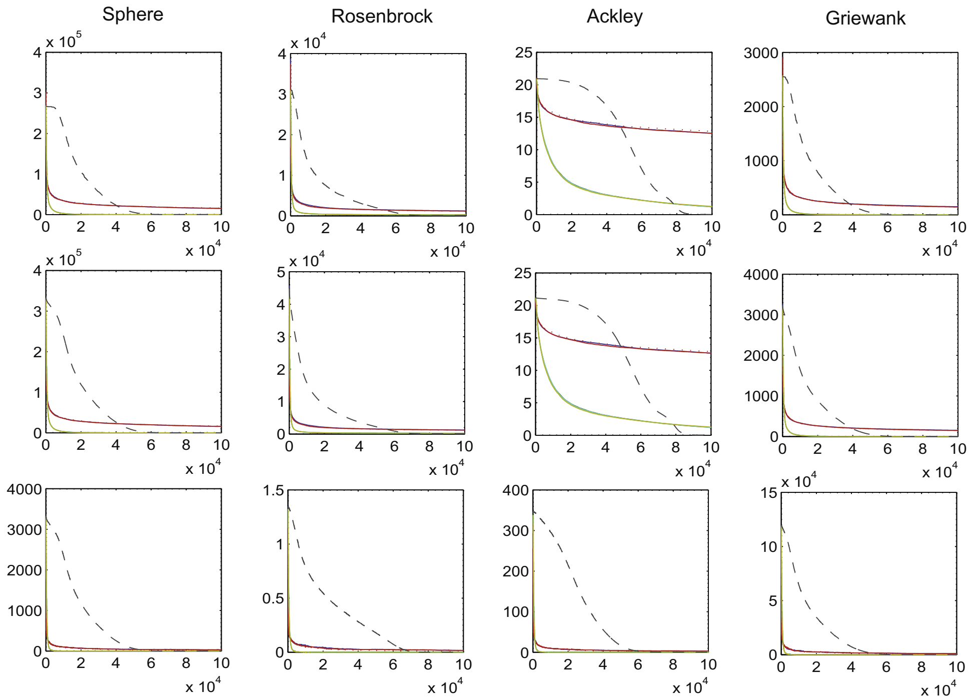


Fig. 9. Comparisons among seven methods with **negative symmetric** initial range on four test functions in **100** dimensions. The dash lines indicate the results of the proposed method. The apparent red lines indicate the results of the original HS, IHS and GHS, and the green lines indicate the HS_{50}, IHS_{50} and GHS_{50}. They are too close to be distinct. Three rows respectively indicate the best result, the average result, and the diversity. The vertical axes of the first two rows and the third row are the function values and the diversity, respectively. The horizontal axis is the number of function evaluations. All results have been averaged over 30 runs.

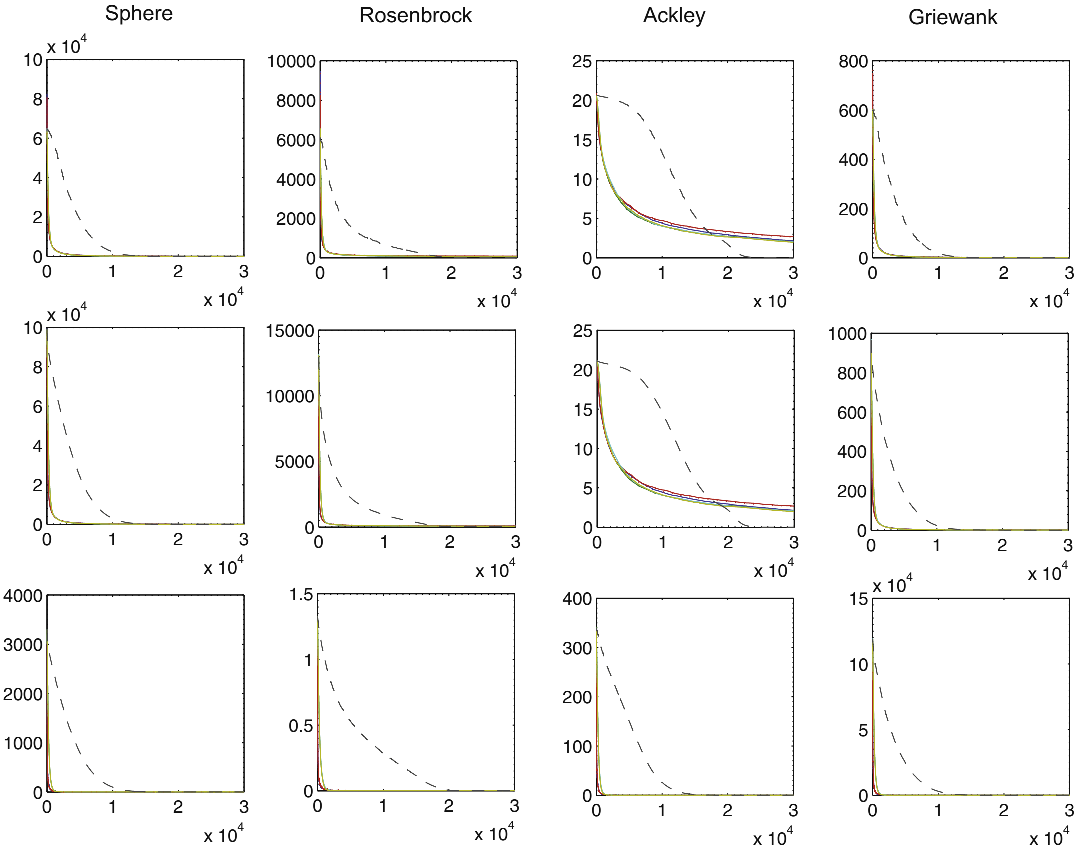


Fig. 10. Comparisons among seven methods with **symmetric** initial range on four test functions in **30** dimensions. The dash lines indicate the results of the proposed method. The apparent red lines indicate the results of the original HS, IHS and GHS, and the green lines indicate the HS_{50}, IHS_{50} and GHS_{50}. They are too close to be distinct. Three rows respectively indicate the best result, the average result, and the diversity. The vertical axes of the first two rows and the third row are the function values and the diversity, respectively. The horizontal axis is the number of function evaluations. All results have been averaged over 30 runs.

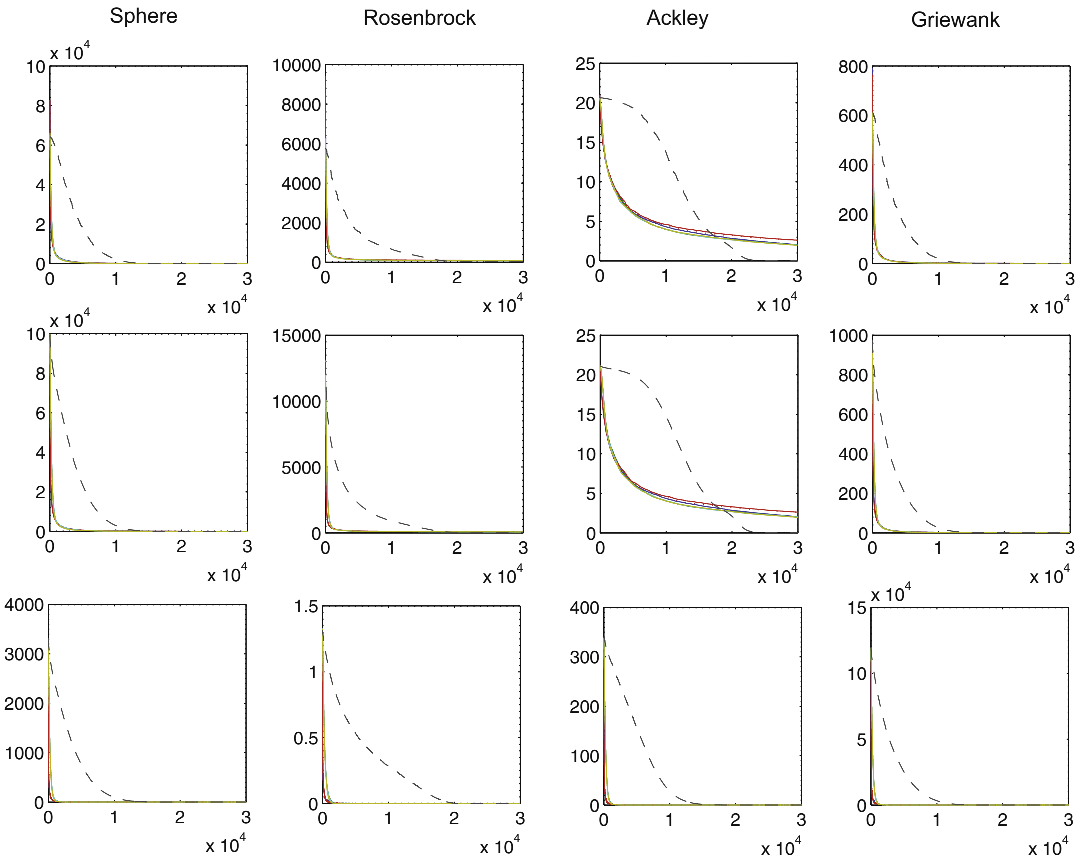


Fig. 11. Comparisons among seven methods with **positive symmetric** initial range on four test functions in **30** dimensions. The dash lines indicate the results of the proposed method. The apparent red lines indicate the results of the original HS, IHS and GHS, and the green lines indicate the HS_{50}, IHS_{50} and GHS_{50}. They are too close to be distinct. Three rows respectively indicate the best result, the average result, and the diversity. The vertical axes of the first two rows and the third row are the function values and the diversity, respectively. The horizontal axis is the number of function evaluations. All results have been averaged over 30 runs.

and Mahdavi (2008), and the parameters for our method were obtained from the observations of factorial experiments. Additionally, in order to ensure a fair comparison, we also raised the HMS and the HMCR of the original HS, IHS and GHS to 50 and 0.99, respectively.

Following the advices of Fogel and Beyer (1995) and Angeline (1998) for those problems with optima at or near the origin, the HM was asymmetrically initialized in the search space. Three initialization methods were used in this paper, and the initial HM of all methods was generated by the low-discrepancy sequence generator in the ranges specified in Table 3.

4.4. Numerical results

The final numerical results averaged over 30 runs in 30 and 100 dimensions are given in Tables 4 and 5. As shown in these tables, the numerical results clearly indicate that the proposed method significantly outperforms all other methods under 99% confidence interval *t*-test in all test functions. From the perspective of initialization ranges, the performances of all methods in all test functions are generally quite consistent. In other words, the harmony search and its variants are not affected by the different initialization ranges. Since the location of the optima in the real-world problem are generally not known, this property guarantees a consistent result if the optima are not included in the initialization range. Besides, it is worth pointing out that the HS_{50}, IHS_{50} and GHS_{50} get better performances than those methods with suggested parameters!

4.5. Analysis of dynamics

In order to realize the dynamic behavior of the harmony search, we defined a diversity measure as follow:

$$\text{Diversity} = \frac{1}{\text{D}} \cdot \sum_{i=1}^{\text{D}} \text{VAR}(\text{HM}^i) \tag{3}$$

where D is the dimension of the HM, and VAR is the variance of each variable (dimension). The total variance of the HM is divided by D to reduce the effect of the dimension.

As shown in Fig. 5, we plotted the learning curve of the best and average of the function values over iterations for all test functions in the first two rows. In addition, the third row is the diversity of the HM across iterations. Although these lines are overlapped, three apparent groups still can be distinguished. It is obvious from the third row that our method preserves a certain degree of diversity, and finally converges to a better solution. However, all other methods have a strong tendency to converge prematurely to relatively poor solutions. Especially for the Ackley function, the original HS, IHS and GHS get stuck in the local optimum very soon, so that they are hopeless to reach a better solution near the global optimum. Despite that the HS_{50}, IHS_{50} and GHS_{50} benefit from the larger HMCR and HMS, they still have a tendency to converge to non-optimal solutions. It is clearly evident by the diversity curve of the Ackley function and the numerical results. In general, our method decreases the function values and the diversity gracefully in all test functions, i.e., it could continuously find better solutions

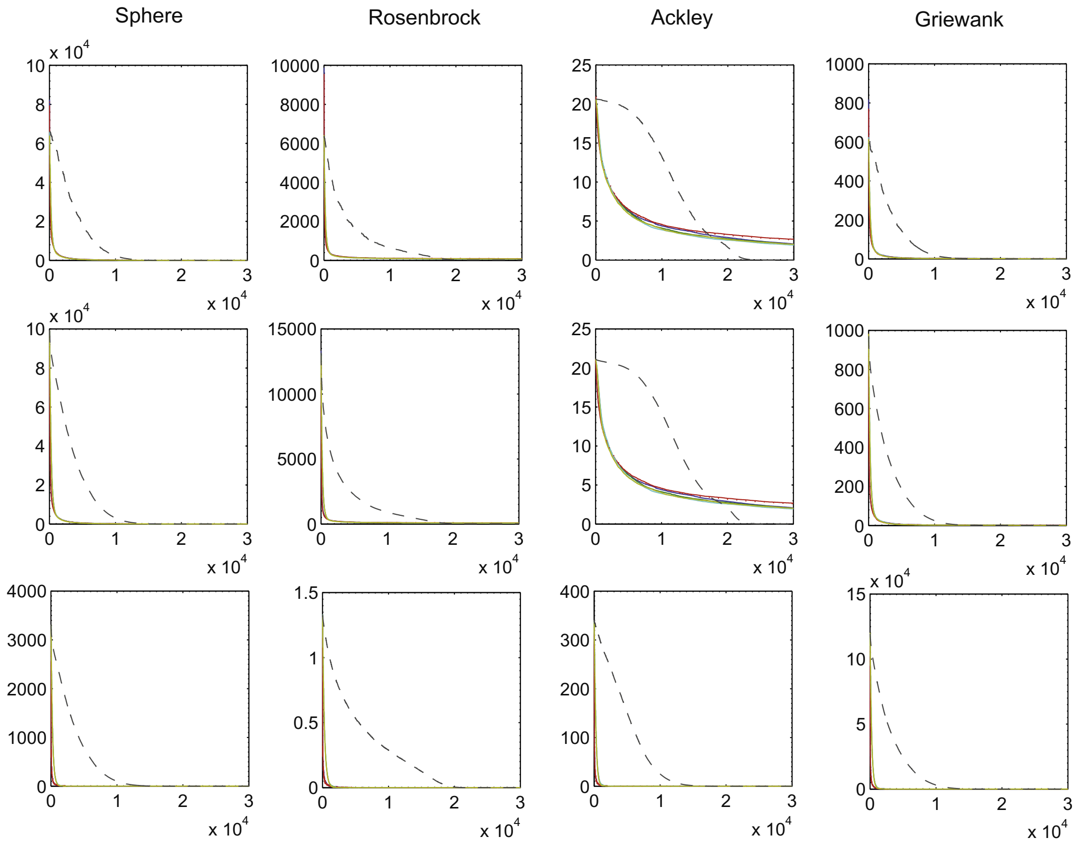


Fig. 12. Comparisons among seven methods with **negative symmetric** initial range on four test functions in **30** dimensions. The dash lines indicate the results of the proposed method. The apparent red lines indicate the results of the original HS, IHS and GHS, and the green lines indicate the HS_{50}, IHS_{50} and GHS_{50}. They are too close to be distinct. Three rows respectively indicate the best result, the average result, and the diversity. The vertical axes of the first two rows and the third row are the function values and the diversity, respectively. The horizontal axis is the number of function evaluations. All results have been averaged over 30 runs.

through the iterations, and guarantee the convergence to a better solution.

5. Conclusions

In this paper, we proposed a new harmony search variant, the parameters in which are automatically adapted based on past experiences. This automatic adjustment mechanism not only alleviates the difficulties of parameter setting, but also enhances the precision of solutions. In addition, we carefully conducted a set of experiments to understand the impact of control parameters. It was observed that a large value of HMCR and a moderate size of harmony memory (i.e., 50) are a suitable set of parameters. The results remained consistent in high dimensional problems.

Based on these observations, the proposed method was compared with the original HS, IHS, and GHS that used two set of parameters; i.e., the default parameters provided by their authors and the parameters obtained from the full-factorial experiments. The numerical results indicated that our method offers much superior performance to the existing methods on four optimization problems. Besides, it is interesting to note that these three methods can get better performance with the parameters obtained from the factorial experiments. This circumstance confirmed our criticisms on existing conclusions and heuristic values of parameters in Section 1. Furthermore, our method in contrast to others has a better diversity and always converges to a better solution eventually. We are currently investigating our method with more difficult problems, especially in higher dimensions.

Appendix A

A.1. Standard test functions

See Fig. 6.

A.2. Design of experiments

See Fig. 7.

A.3. Numerical results

See Tables 6 and 7.

A.4. Analysis of dynamics

See Figs. 8–12.

References

Angeline, P. J. (1998). Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In *EP '98: Proceedings of the 7th international conference on evolutionary programming VII* (pp. 601–610). London, UK: Springer-Verlag.

Czitrom, V. (1999). One-factor-at-a-time versus designed experiments. *The American Statistician, 53*(2), 126–131.

Fogel, D. B., & Beyer, H.-G. (1995). A note on the empirical evaluation of intermediate recombination. *Evolution Computing, 3*(4), 491–495.

Geem, Z. (2007). Harmony search algorithm for solving sudoku. In *Knowledge-based intelligent information and engineering systems of lecture notes in computer science* (Vol. 4692, pp. 371–378).

Geem, Z. W. (2008). Harmony search applications in industry. *Soft Computing Applications in Industry, 226*, 117–134.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *SIMULATION, 76*(2), 60–68.

Glover, F. (1989). Tabu search – Part I. *ORSA Journal on Computing, 1*(3), 190–206.

Haykin, S. (2007). *Neural networks: A comprehensive foundation* (3rd ed.). Upper Saddle River, NJ, USA: Prentice-Hall, Inc..

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680 (Number 4598, 13 May 1983).

Lecot, C. (1989). An algorithm for generating low discrepancy sequences on vector computers. *Parallel Computing, 11*(1), 113–116.

Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering, 194*(36–38), 3902–3933.

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation, 188*, 1567–1579.

Matsumoto, M., & Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling Computer Simulation, 8*(1), 3–30.

Montgomery, D. C., & Runger, G. C. (2006). *Applied statistics and probability for engineers*. Wiley.

Omran, M. G., & Mahdavi, M. (2008). Global-best harmony search. *Applied Mathematics and Computation, 198*(2), 643–656.

Pardalos, P. M., & Resende, M. G. C. (Eds.). (2002). *Handbook of applied optimization*. Oxford University Press.